

Pokémon Battle Predictor

Mercurio Raffaele
1883984

Salvatore Vincenzo
1918339

Sforza Luca
2050030

1. Introduction

Predicting Pokémon battle outcomes is a complex task due to team interactions and strategic variability. In this project, we developed models to predict the winning team after a fixed number of turns, applying a structured data engineering process and testing various machine learning and ensemble methods to maximize accuracy.

2. Data Engineering

2.1. Preliminary Data Analysis

An exploratory analysis was first conducted to understand the structure, completeness and consistency of the data. Each sample contains the information of the first player's team and the lead Pokémon of the second player. It also contains the first 30 turns of the battle, where we have desumed the other Pokémons of the second team and the modified stats for every Pokémon involved. To ensure data consistency and facilitate efficient querying and preprocessing, all battle data were imported into a relational database.

2.2. Feature Selection and Aggregation

Given the complexity and granularity of the original data (e.g., individual moves, temporary status effects and ability activations), we opted to focus on a set of aggregate features that could effectively capture the overall performance of each team. Specifically, we decided to preserve only the final statistics of the Pokémons belonging to each team after the 30th turn (i.e., Final HP, Attack, Defense, Special Attack and Defense, Speed, Boosts, Type, Status). Some features were not in numerical form (e.g., Type and Status), so we encoded them using a *MultiLabelBinarizer* for Pokémon types (since they can have up to two) and a *OneHotEncoder* for Pokémon status (since they can have only one state at a time). Eventually, for each battle, the statistics of all six Pokémon belonging to the same team were aggregated using summary measures, resulting in a single feature vector per team. Subsequently, the feature vector of the second team was subtracted from that of the first, producing a single representation that captures the difference in strength and composition between the two teams. Finally,

a PCA was applied to the resulting data, retaining only the features that explained 99% of the variance: *Final HP, Attack, Defense, Special Attack, Special Defense, Speed, No Status, Status Fainted and Paralyzed, Psychic, Normal, Water Grass, Rock, Ground, Flying, Poison Types*. This approach significantly reduced the dimensionality of the data while preserving the most relevant information for predicting battle outcomes.

3. Models Used

3.1. Models and Parameters

After that, a *model selection* process was carried out by training and optimizing several machine learning models to predict the outcome of each Pokémon battle. Hyperparameter tuning was performed through a combination of grid and randomized search strategies, using validation accuracy as the main performance metric. The best configurations and their corresponding accuracies are summarized in Table. 1.

Table 1. Best parameters and accuracy for each model.

Model	Best Parameters	Accuracy
KNN	n_neighbors=28, weights=distance, metric=minkowski (p=2)	0.8193
Logistic Regression	solver=lbfgs, penalty=l2, max_iter=10000, Cs=10, cv=5, n_jobs=8	0.8405
Decision Tree	criterion=gini, max_depth=6, splitter=best	0.8152
Random Forest	criterion=entropy, n_estimators=100, max_depth=97, max_features=sqrt	0.8304
XGBoost	objective=binary:logistic, max_depth=6, reg_lambda=0.47	0.8290

3.2. Ensemble

Eventually, both an ensemble model and a stacking meta-model were implemented using the same five base learners (Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest and XGBoost). The ensemble relied on majority voting among the individual predictions, while the stacking model combined them through a Logistic Regression meta-learner, achieving a more stable and robust performance.